LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-416785

DAPA-V10: Discovery and Analysis of Patterns and Anomalies in Volatile Time-Evolving Networks

B. Thompson, T. Eliassi-Rad

September 14, 2009

The 1st Workshop on Information in Networks
New York, NY, United States
September 25, 2009 through September 26, 2009

Disclaimer

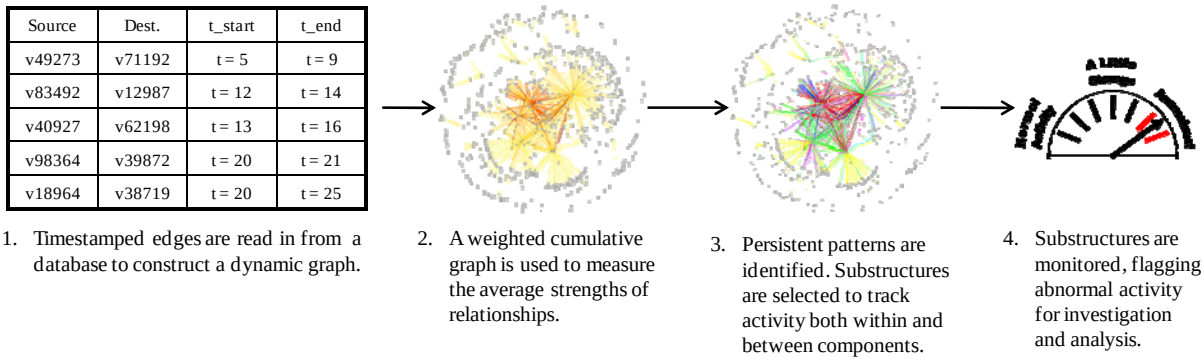
DAPA-V10: Discovery and Analysis of Patterns and Anomalies in Volatile Time-Evolving Networks*

Brian Thompson
Rutgers University
bthom@cs.rutgers.edu

Tina Eliassi-Rad
Lawrence Livermore National Laboratory
eliassirad1@llnl.gov

Introduction

We address the problems of finding patterns and detecting anomalous activities in *volatile* time-evolving networks such as communication networks (as opposed to slowly evolving networks like co-authorship graphs). Our approach, *DAPA-V10*, utilizes a simple compact graph representation that assigns weights to edges in a way that captures the frequency, duration, and recency of edges. Given this weighted "cumulative" graph, DAPA-V10 finds *persistent patterns* by extracting connected components of regularly-occurring edges. These persistent patterns provide a basis for expected normal behavior in the network over time, which are then utilized to detect anomalous behavior on both local and global scales. In particular, DAPA-V10 uses a scalable approach based on the Product Rule for the Central Limit Theorem to measure the likelihood of events and flag anomalies. Figure 1 provides an overview of DAPA-V10. Experiments on the Enron email dataset illustrate the effectiveness of our approach.



Source	Dest.	t_start	t_end
v49273	v71192	t = 5	t = 9
v83492	v12987	t = 12	t = 14
v40927	v62198	t = 13	t = 16
v98364	v39872	t = 20	t = 21
v18964	v38719	t = 20	t = 25

1. Timestamped edges are read in from a database to construct a dynamic graph.
2. A weighted cumulative graph is used to measure the average strengths of relationships.
3. Persistent patterns are identified. Substructures are selected to track activity both within and between components.
4. Substructures are monitored, flagging abnormal activity for investigation and analysis.

Figure 1. Pictorial overview of DAPA-V10. The graphs in steps 2 and 3 are from the Enron dataset with darker edges representing stronger relationships.

DAPA-V10: Graph Representation

We model a time-evolving network as a dynamic graph $G = (V, E_T)$, composed of a fixed set of vertices V and a set of time-stamped edges E_T. The graph $G_t = (V, E_t)$ represents the network at time t. For simplicity, we assume that G_t is not a multi-graph (i.e., there is at most one edge between each pair of vertices at any given time.) To capture recent activity, we construct a weighted *cumulative graph*, $G'_t = (V, E'_t, W'_t)$. It encapsulates all past edges but gives greater weight to more recent ones. We formalize the cumulative graph as follows. Given a dynamic graph G and a decay function f, the cumulative graph at time t is the weighted graph G'_t where each edge e has weight:

$$w(e) = \sum_{e \in E_T} f(e)$$

* This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344; and partially supported by the U.S. Department of Homeland Security under Grant Award Number 2008-ST-104-000016. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S Department of Homeland Security.

The decay function $f(e)$ may depend on the semantics, duration, frequency, and recency of the edges. A popular choice for time-evolving networks is the exponential decay model, but other models may be used as appropriate for a given dataset.

DAPA-V10: Identifying Persistent Patterns

In order to recognize an anomalous event, DAPA-V10 first establishes a basis for normal behavior by identifying *persistent patterns* among vertices. A persistent pattern is a collection of vertices that (1) form a connected component and (2) communicate regularly. Given a cumulative graph G', we can identify persistent patterns by extracting connected components composed of edges whose weights are above a threshold θ. In full generality, the value of θ may be different for each component, and choosing appropriate values may depend on multiple factors such as the distribution of the edge weights and the semantics associated with the edges. The intention is that a persistent pattern represents a set of vertices with regular communication patterns, which can subsequently be used to detect deviations from the norm.

In this work, we utilize the following threshold scheme. First, we generate the *average-weight graph* by assigning to each edge a weight corresponding to the fraction of time that edge exists in the graph. We then use a sliding threshold to extract components from the average-weight graph. In this scheme, the weight threshold is gradually decreased until a component of size greater than square root of $|V|$ emerges. Edges in the component are removed from the graph, and the process is then repeated on the remaining graph. Note that since only edges are removed and not vertices, a single vertex may be in several persistent patterns. This is a realistic model since people typically engage in an assortment of communications. Figure 2 depicts the distribution of edge weights from the average-weight Enron email graph, along with the threshold points used to determine components. The persistent patterns found in the Enron email graph are highlighted in Figure 1 (Step 3).

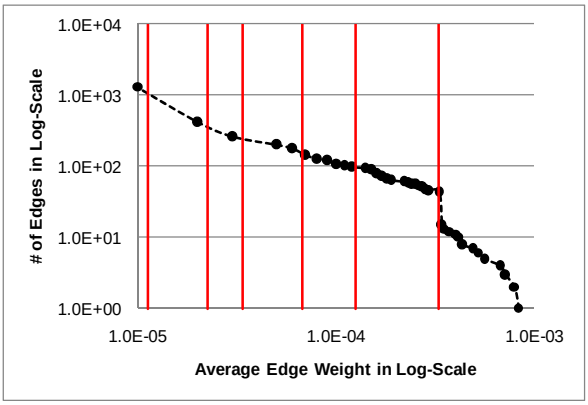


Figure 2. Cumulative distribution of edge weights in the average-weight Enron email graph. The red vertical lines represent threshold points (computed as described above). From left to right, they are: 1.1E-05, 2.3E-05, 3.4E-05, 6.8E-05, 1.3E-04, and 3.3E-04.

DAPA-V10: Anomaly Detection

To detect anomalies, we compare current activity at a particular time with the expected activity based on recent behavioral patterns, and classify an event as anomalous if it differs significantly from the expected activity. We model recent behavior using the aforementioned cumulative graph with exponential decay and a half-life of 10 days.

Our goal is to identify *any anomalous local or global event*. More specifically, we want to analyze whether a subset of vertices in the network $U \subseteq V$ is exhibiting anomalous behavior at time t. Let w_e denote the weight of edge e in the cumulative graph G'_t. For each edge e, we define the random variable: $X_e = \{w_e$ if e in E_t, $1 - w_e$ otherwise$\}$. To measure likelihood of events (determined by which edges in U exist at time t), we define a product random variable:

$$X_U = \prod_{i,j \in U} X_{(i,j)}$$

Values of X_U provide our likelihood measure.

To determine whether a subset U is anomalous at time t, we compare the value $X_U(t)$ to the distribution of values of X_U, which can be approximated by appealing to the Product Rule for the Central Limit Theorem:

$$D_{X_U} \sim LogNormal\left(\mu_U^* = \sum \mu_e^*, \sigma_U^* = rms\left(\sigma_e^*\right)\right)$$

where μ_e^* and σ_e^* are the mean and standard deviation, respectively, of the random variable X_e^* = **log** X_e. We approximate the distribution of X_e by using p_e = (# of occurrences of e before time t)/t as an estimate of the probability that edge e exists at time t. The subset U is identified to be anomalous at time t if

$$\frac{\left|X_U(t) - \mu_U^*\right|}{\sigma_U^*} > \alpha$$

where α is a pre-defined *anomalicity threshold*.

Alternatively, one may consider DAPA-V10 as assigning an *anomalicity score* to every subset being analyzed. In practice this may be used to flag the most anomalous events for further investigation, proceeding in decreasing order of anomalicity as time and resources allow.

Our approach can be used to monitor a given substructure of the network for anomalous behavior, but does not give an intuition for which substructures to monitor. DAPA-V10 combines this technique with the persistent pattern identification algorithm described in the previous section. Specifically, we independently monitor sets of edges within each identified persistent pattern and between each pair of patterns, as well as the edges incident to each single vertex. DAPA-V10 is not restricted to these substructures, however; others may be selected for monitoring as desired, and may depend on the application.

Recall that a single vertex can belong to multiple persistent patterns depending on the weights on its edges and the threshold scheme. However, since each edge appears in exactly one persistent pattern, the runtime of the DAPA-V10 anomaly detection algorithm is $O(m)$, where m is the number of distinct edges in the network. Therefore, our approach is scalable to very large networks with millions of edges.

Experiments

We test the effectiveness of the DAPA-V10 algorithm using the Enron dataset, a collection of emails sent between Enron employees over a period of 5 years from 1997-2002 (available at http://www.cs.cmu.edu/~enron/). Our goal is to identify persistent patterns in communication among sets of Enron employees and detect anomalous behavior in their email activity. We first clean the raw data by removing emails sent outside of Enron and low-degree vertices, which leaves us with 672 employees and 4417 emails between them.

Using the average-weight graph, our algorithm first finds persistent patterns in the data. Following the threshold scheme detailed above, DAPA-V10 finds 6 subsets of sizes {41, 25, 43, 58, 89, 361} that represent connected components of Enron employees with regular communication. The largest component is composed of many "fringe" vertices that are loosely connected to some of the "core" vertices and occasionally to each other. This observation was also made by Leskovec, et al. [8].

DAPA-V10 then selects a set of substructures in the network, based on the persistent patterns found, and monitors them for anomalous activity as the network evolves. Specifically, it checks whether the activity in the past day differs significantly from the typical behavior over the last couple of weeks (using an exponential decay model with decay rate of 10 days). In our experiments, we track three types of substructures: vertex neighborhood structures (email of a single employee), intra-link structures (email between employees in a persistent pattern), and inter-link structures (email between employees in two separate persistent patterns).

Figure 3 compares anomalies found by DAPA-V10 with events surrounding the Enron scandal. The close correspondence illustrates the effectiveness of our approach. The full timeline of the Enron scandal is available at http://en.wikipedia.org/wiki/Timeline_of_the_Enron_scandal#1999. It includes details about the anomalies found prior to 2001. For example, the anomaly discovered at the end of 1999 corresponds to the launch of EnronOnline.

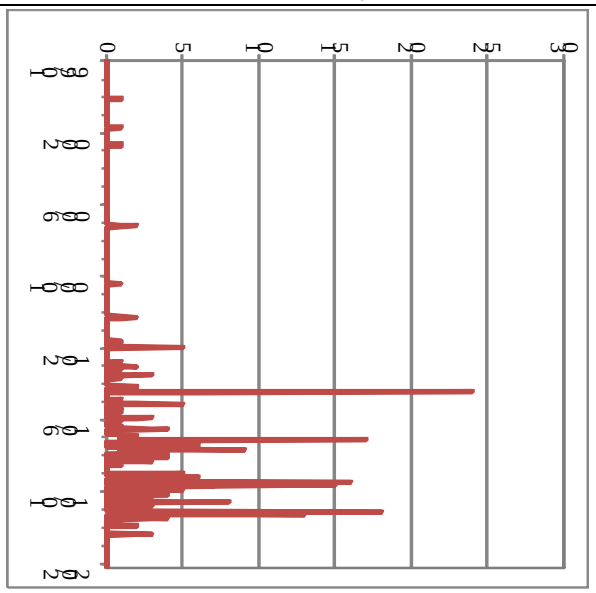
Timeline of Enron Scandal		# of Anomalies Found by DAPA-V10
Time	Event	
2/01	Executives get $1M bonuses; stock is soaring	
4/01	Q1 profit $536M; Wall St. analyst suspicious	
7/01	Reported earnings $50B; share price dropping	
8/01	Public criticism of Enron accounting practices	
9/01	9/11 attacks; Enron director sells 500K shares	
10/01	Q3 loss of $618M; SEC begins investigation	
11/01	Acquisition offer, revoked; 'junk' credit rating	
12/01	Enron files for bankruptcy, lays off employees	

Figure 3. The number of anomalies detected by the DAPA-V10 algorithm on the Enron email dataset from October 1999 to February 2002, juxtaposed with a timeline of events surrounding the Enron scandal.

Related Work

Akoglu *et al*. [1] provide a nice empirical study of laws governing weighted time-evolving graphs. Goetz *et al*. [4] focus on blogs and model the bursty behavior observed there. Sharan and Neville [10] use a weighted *summary graph* (with an exponential decay model) that is similar to our cumulative graph. However, their task is to construct temporal-relational classifiers for time-evolving graphs. Kahanda and Neville [5] describe a statistical relational learning approach for predicting link strength in online social networks. DAPA-V10 can utilize their approach to help estimate edge strengths when identifying persistent patterns.

Our persistent patterns can also be thought of as communities in a social context. Community discovery in time-evolving graphs has been given a good deal of attention lately [6, 7, 9]. Much of this work defines communities as a clustering of the vertices in a graph with high modularity (high intra-cluster and low inter-cluster density), or by using compression-based techniques [11]. Our persistent patterns provide a more general definition that allows for more variety in communication patterns.

Sun, *et al.* [11] developed an anomaly detection algorithm called GraphScope, which finds communities in a dynamic graph by utilizing a compression-based approach. A comparison of the anomalies detected by DAPA-V10 and GraphScope on the Enron dataset shows that they find many of the same anomalies. However, there are several differences between our approach and GraphScope. For instance, GraphScope does not permit overlapping communities, while our definition of persistent patterns allows vertices to be members of multiple communities.

Lastly, there has been some work on identifying times of heightened activity across an entire network [11, 12]. However, as mentioned before, DAPA-V10 is able to identify anomalous events on both a local and global scale. Chandola, *et al*. [3] have a nice recent survey on anomaly detection. Boettcher, *et al*. [2] provide an excellent overview of *change mining*.

Future Work

As our next step, we intend to conduct a comprehensive set of experiments on various volatile time-evolving networks. Our goal here will be to identify exact substructures that are correlated with anomalous behavior. Of particular interest is identifying subtle activities that may not be visible on a global scale.

Another direction is to study edge correlations, that is, edges whose patterns of occurrence are not independent. This could affect how we determine an anomalicity score for each substructure, and may help eliminate false positives. We could also normalize edge weights at each time step based on the total network activity at the time. This would help to find local anomalies independent of global trends in network activity.

Incorporating semantic information from complex networks is another promising direction of future work. Many real-life networks are rich in semantic information, which could greatly improve the accuracy and reliability of DAPA-V10 in identifying anomalous activity.

Conclusions

We present DAPA-V10, a novel algorithm that addresses the task of anomaly detection in time-evolving networks. Our approach first identifies persistent patterns in the network. For volatile time-evolving networks, such as many communication networks, the task of identifying persistent patterns in the data is non-trivial and important in its own right. Then, based on those patterns, we select subgraphs to monitor for anomalous activity at the network evolves.

One major advantage of DAPA-V10 is that it can find *local* anomalies efficiently, whereas previous work focuses mainly on identifying global anomalies (i.e., identifying times of higher activity levels overall). This advantage allows us to pinpoint the source of unusual behavior for further analysis. Furthermore, by assigning an anomalicity score to each substructure, they can be ranked in order of anomalicity. In this way, DAPA-V10 can serve as a guide for human-directed analysis, where analysts may choose to investigate as few or as many potentially anomalous substructures as they deem necessary.

Finally, we evaluate our algorithm by running experiments on the Enron email dataset. With the growing security challenges faced today, from monitoring network traffic for spam or viruses to exposing botnets, DAPA-V10 provides a promising avenue for an effective solution to the problem of anomaly detection in volatile time-evolving networks.

References

[1] L. Akoglu. M. McGlohon, and C. Faloutsos. RTM: Laws and a Recursive Generator for Weighted Time-Evolving Graphs. In ICDM'08, pp. 701-706, 2008.

[2] M. Boettcher, F. Hoeppner, and M. Spiliopoulou. On Exploiting the Power of Time in Data Mining. SIGKDD Explorations 10(2), pp. 3-11, 2008.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. In ACM Computing Surveys 41(3), Article 15, July 2009.

[4] M. Goetz, J. Leskovec, M. McGlohon, and C. Faloutsos. Modeling Blog Dynamics. In ICWSM'09, San Jose, CA, May 2009.

[5] I. Kahanda and J. Neville. Using Transactional Information to Predict Link Strength in Online Social Networks. In ICWSM'09, San Jose, CA, May 2009.

[6] B. Karrer, E. Levina, M. E. J. Newman. Robustness of Community Structure in Networks. Physical Review E, 2008.

[7] E. A. Leicht, G. Clarkson, K. Shedden, M. E. J. Newman. Large-Scale Structure of Time-Evolving Citation Networks. In European Journal of Physics, 2007.

[8] L. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Statistical Properties of Community Structure in Large Social and Information Networks. In WWW'08, pp. 695-704, 2008.

[9] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying Social Group Evolution, Nature 446(7136), pp. 664-667, 2007.

[10] U. Sharan and J. Neville. Temporal-Relational Classifiers for Prediction in Evolving Domains. In ICDM'08, pp. 540-549, 2008.

[11] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. GraphScope: Parameter-Free Mining of Large Time-Evolving Graphs. In KDD'07, pp. 687–696, 2007.

[12] H. Tong, Y. Sakurai, T. Eliassi-Rad, and C. Faloutsos. Fast Mining of Complex Time-Stamped Events. In CIKM'08, pp. 759-768, 2008.